REMARKS

The above-identified application is United States application serial number 10/037,593 filed on October 19, 2001. Claims 1-3, 5-19 and 21-27 are pending in the application. Claims 1-3, 5-19 and 21-27 are rejected.

Claim Rejections under 35 U.S.C. §102

Claims 1-3, 5-19, 21-27 are rejected under 35 U.S.C. 102(b) as being anticipated by Rustad *et al.* US Patent 6,643,717 B1 (hereinafter "Rustad"). Claim 1 recites:

> "a Regional Transaction Processor (RTP); and
> a Data Enabling Device (DED) operable to:
> receive one or more data packets from the information network,
> detect when the one or more data packets include content match information, and
> issue a message to a workstation and invoke the RTP to process a transaction when the content match information is detected in the one or more data packets, wherein the DED is operable to prevent further transmission of the one or more data packets based on the content match information."

Rustad discloses flow control for a serial link that includes setting a bit in a hardware register when a match is detected. (Rustad col. 3 lines 54-55.) The bit register is associated with the serial link. (Rustad col. 3 lines 36-42.) Rustad does not teach or suggest issuing a message to a workstation when the content match information is detected in one or more data packets. Rustad further does not disclose or suggest a component that is equivalent to a workstation. Further, Rustad does not disclose or suggest a component that is equivalent to the RTP to process a transaction, or even a transaction that is processed. If preventing further transmission of the data packets is considered to be a transaction, then Rustad does not disclose or suggest both controlling transmission of the packets AND processing a transaction as set fort in Claim 1. Claim 1 is allowable over Rustad for at least these reasons.

Claims 2-3 and 5-17 depend from Claim 1 and include features that further distinguish them from the cited reference. For example, Claim 5 recites "wherein the RTP comprises a network server and a database, and is operable to process transactions for requests for content." In contrast, the cited portion of Rustad teaches a serial port card that includes an interface for a plurality of serial ports, FIFO buffers, a direct memory access controller, a local bus, and a processor. (Rustad col. 2 lines 49-60). There is no teaching or suggestion that the DMA controller, FIFO buffers, local bus, and/or the processor in Rustad are equivalent to a network server or database, or perform the function of processing transactions for requests for content in addition to controlling data flow. Rustad thus does not teach or suggest a network server, a database, or processing transactions for requests for content. Claim 5 is allowable over Rustad for at least these additional reasons.

As another example, claim 7 recites "a plurality of DEDS along a network route, wherein each DED is operable to communicate with at least one of the other DEDS." In contrast, the cited portion of Rustad teaches "[w]hen the bit used for flow control is set, it indicates that the associated serial port has received data that may be a command to cease transmissions. This bit can be set under hardware or software control." (Rustad column 3, lines 6-9). Thus, the cited portion of Rustad does not teach or suggest a plurality of DEDs along a network route that are operable to communicate with at least one of the other DEDs. Claim 7 is allowable over Rustad for at least these additional reasons.

As another example, claim 8 recites "wherein the plurality of DEDS include a first DED that generates a message and one or more intermediate DEDS operable to forward the message to the DED closest to the workstation along the network route." The cited portion of Rustad teaches a process for flow control for a serial link that includes a bit register that is set when a group of bits received on the serial link matches a selected bit pattern. (Rustad column 3, lines 35-40). Rustad does not teach a DED that generates a message or intermediate DEDs operable to forward the message to the DED closest to the workstation along the network route. Claim 8 is allowable over Rustad for at least these additional reasons.

As another example, claim 9 discloses a plurality of DEDS operable to communicate with each other to prevent transmitting more than one message for the same data packet through the network route. In contrast, the cited portion of Rustad teaches "if...the data was a command, then the method proceeds to ... determine the effect, if any, that the command has on the flow control state. [T]he method determines whether the command is a stop command." (Rustad column 4, lines 23-27). Rustad thus does not teach DEDs communicating with each other to prevent transmitting more than one message for the same data packet through the network route. Claim 9 is allowable over Rustad for at least these additional reasons.

As another example, claim 11 discloses FPGAS that can be reprogrammed over the network to perform a content matching function. The cited portion of Rustad describes a process for flow control for a serial link that includes a bit register that is set when a group of bits received on the serial link matches a selected bit pattern. (Rustad column 3, lines 35-42). Rustad further discloses monitoring incoming data on a serial link for bits that match a selected bit pattern corresponding to a selected command. (Rustad column 3, lines 43-46). Although Rustad further states that other techniques can be used to detect a match between data in the data stream and the selected command, Rustad does not disclose or suggest reprogramming the bit pattern over a network, as recited in claim 11. Claim 11 is allowable over Rustad for at least these additional reasons.

As a further example, claim 13 discloses that a portion of the DED can be dynamically reprogrammed and the DED continues processing the data packets during the partial reprogramming. The cited portion of Rustad teaches setting a flow control state based on whether the detected command is a STOP command or a START command. (Rustad column 4, lines 30-43). The cited portion of Rustad thus does not disclose or suggest dynamically reprogramming the DED while continuing to process the data packets as recited in claim 13. Claim 13 is allowable over Rustad for at least these additional reasons.

As yet another example, claim 14 includes a Central Storage and Backup System (CSBS) operable to communicate with the RTP, to monitor operation of the RTP, and to store transaction information. The cited portion of Rustad teaches a method that monitors incoming data on the serial link for bits that match a selected bit pattern corresponding to a selected command, e.g., the "XOFF" command or its equivalent. (Rustad column 3, lines 44-46). The Examiner has not stated which component in Rustad is equivalent to the CSBS of claim 14. When Claim 14 is combined with the features of claim 1, the system includes an

RTP to process a transaction as well as a CSBS to monitor operation of the RTP and store transaction information. Applicant submits that these features are not taught or suggested by the cited portion of Rustad. Claim 14 is allowable over Rustad for at least these additional reasons.

Claim 15 recites that the CSBS is operable to transmit information to reprogram the DED to communicate with another RTP. The cited portion of Rustad teaches determining that the command is not a start command, and then whether there is additional data. If so, Rustad processes the next data. Rustad then determines the current flow control state for the serial port based on the state of the STATE variable. (Rustad column 4, lines 35-43). Applicant submits that setting a state variable based on whether a command is a start command in Rustad is the same as transmitting information to reprogram the DED to communicate with another RTP as recited in Claim 15. Claim 15 is allowable over Rustad for at least these additional reasons.

Claim 16 recites a content matching server operable to store content match information, to communicate with the DED, and to transmit the content match information to the DED. The cited portion of Rustad discloses "[a]t block 316, the method determines the current flow control state for the serial port based on the state of the STATE variable. If the STATE variable indicates a STOPPED state, then the method ends at block 320. If the STATE variable indicates a NOT_STOPPED state, then the method restarts transmissions at block 318 and ends at block 320." (Rustad column 4, lines 41-46). The state variable is set based on whether the incoming data includes bits that match a selected bit pattern corresponding to a selected command. (Rustad column 3 lines 44-47). Nothing in Rustad discloses or suggests that the bit pattern to be matched (content match information) is transmitted to the serial port. Rather, the serial port in Rustad includes a match register that is pre-programmed with a bit pattern, and bits in the data transmitted to the port are compared to the contents of the match register(s). (Rustad column 4 line 61 through column 5 line 37). Claim 16 is allowable over Rustad for at least these additional reasons.

Claim 17 recites "the DED is operable to suspend transmission of the data packets through the information network until a response to a prompt is received." The cited portion of Rustad teaches a process for flow control for a serial link including a bit register that is set when a group of bits received on the serial link matches a selected bit pattern. (Rustad column 3, lines 35-41). Rustad does not disclose or suggest suspending transmission of the data packets until receiving a response to a prompt. Claim 17 is allowable over Rustad for at least these additional reasons.

Claims 18, 26, and 27 include "[i]ssuing a prompt to a workstation based on the content match information when the content match information is detected in the at least one data packet." The cited portion of Rustad teaches "[i]f, at block 304, the data was a command, then the method proceeds to block 306 to determine the effect, if any, that the command has on the flow control state. At block 306, the method determines whether the command is a stop command. If it is a stop command, the method proceeds to block 308 and sets a STATE variable to a STOPPED state." (Rustad column 4 lines 23-33). Applicant submits that setting a variable is not equivalent to issuing a prompt. Further Rustad does not disclose or suggest sending the variable to a workstation. Claims 18, 26, and 27 are allowable over Rustad for at least these reasons.

Claims 19 and 21-25 depend from Claim 18 and include features that further distinguish them from the cited reference. For example, claim 21 recites processing a transaction based on a user's response to the prompt. The cited portion of Rustad teaches determining "the current flow control state for the serial port based on the state of the STATE variable. If the STATE variable indicates a STOPPED state, then the method ends at block 320. If the STATE variable indicates a NOT_STOPPED state, then the method restarts transmissions at block 318 and ends at block 320." (Rustad column 4, lines 40-46). Rustad does not include processing a transaction based on a user's response to the prompt. Claim 21 is allowable over Rustad for at least these additional reasons.

Claim 22 recites transmitting a message among a plurality of DEDS along the transmission path to prevent transmitting more than one prompt for the same data packet. The cited portion of Rustad teaches "[s]ystems and methods have been described that use a combination of hardware and software to set and reset a bit in a register to control the transmission of data from a serial port. Essentially, the hardware aspect allows the system to quickly stop transmission when a bit sequence matches a stop command and the software aspect allows the data stream to be checked to determine the actual flow control state for the serial port. Advantageously, this allows transmissions to be stopped in as little as one character time. If transmissions were inadvertently stopped, the software quickly restarts transmissions." (Rustad column 4, lines 50-60). Rustad only sets a bit in a register in the same serial port where the data is received. The cited portion of Rustad thus does not disclose or suggest "transmitting a message among a plurality of DEDS along the transmission path to prevent transmitting more than one prompt for the same data packet."

Claim 24 recites "reprogramming a portion of the DED to detect different content match information. The cited portion of Rustad teaches "[i]n one embodiment, the method stops transmitting data over the serial link when the bit is set. The method returns to block 202 and continues to monitor the incoming data. The method of FIG. 2 can terminate in one of at least two ways. First, the method ends if the port that is being monitored is closed. Second, the method ends if the port that is being monitored is reconfigured to not use in-band flow control." (Rustad column 3, lines 60-67). Rustad does not disclose or suggest reprogramming a portion of the DED to detect different content match information, as recited in claim 24. When the port is reconfigured in Rustad, the port no longer checks for content match information (i.e., the method of Fig. 2 terminates). Applicant submits that reprogramming content match information is distinguishable from terminating a content match process. Claim 24 is allowable over Rustad for at least these additional reasons.

Claim 25 recites "suspending transmission of the at least one data packet through the information network until a response to the prompt is received." The cited portion of Rustad teaches determining "whether the command is a stop command. If it is a stop command, the method proceeds to block 308 and sets a STATE variable to a STOPPED state. If, at block 306, the method determines that the command is not a stop command, then the method proceeds to block 310 and determines whether the command is a start command." (Rustad column 4, lines 25-33). Applicant submits that suspending transmission of the data packets until receiving a response to a prompt is distinguishable from setting a STATE variable to a STOPPED state if a stop command is detected in a data stream as disclosed by Rustad. Claim 25 is allowable over Rustad for at least these additional reasons.

CONCLUSION

Claims 28-53 have been canceled. Applicants believe the application, including Claims 1-3, 5-19 and 21-27, is in form for allowance and a notice to that effect is solicited. In the event it would facilitate prosecution of this application, the Examiner is invited to telephone the undersigned at (949) 350-7301.

I hereby certify that this correspondence is being transmitted to the USPTO on the date shown below:

/Mary Jo Bertani/
(Signature)

Mary Jo Bertani
(Printed Name of Person Signing Certificate)

September 15, 2006
(Date)

Respectfully submitted,

/Mary Jo Bertani/

Mary Jo Bertani
Attorney for Applicant(s)
Reg. No. 42,321